IOT BASED SMART CROP PROTECTION SYSTEM FOR AGRICULTURE

ORO

S

PET ENGINEERING COLLEGE

An Autonomous Institution
Accredited by NAAC with 'B' Grade
Affiliated to Anna University
Approved by AICTE
Vallioor,
Tamil Nadu-627117

In partial fulfillment for the award of the degree

BACHELOR OF ENGINEERINGING

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
NOVEMBER 2022

A PROJECT REPORT

Submitted by

NAGALAKSHMI A(Team leader) (RollNo:963219106008)

ANTO PANI JONISHA P (RollNo:963219106001)

NANTHINI E (RollNo:963219106009)

SOBIYA N (Roll No:963219106013)

PROJECT REPORT

7. CODING & SOLUTIONING

7.1 Feature1

7.2 Feature2

7.3 Database Schema (if Applicable)

8. TESTING

8.1 Test Cases

8.2 User Acceptance Testing

9. RESULTS

9.1 Performance Metrics

10. ADVANTAGES &DISADVANTAGES

11. CONCLUSION

12. FUTURE SCOPE

13. APPENDIX

1. INTRODUCTION

1.1 PROJECT OVERVIEW

- This project is based on Internet of Things (IoT), that can measure soil moisture, humidity and temperature conditions for agriculture and crop protection using Watson IoT services. IoT is network that connects physical objects or things embedded with electronics, software and sensors through network connectivity that collects and transfers data using cloud for communication. Data is transferred through internet without human to human or human to computer interaction.
- In this project we have not used any hardware. Instead of real soil moisture, humidity and temperature data obtained from sensors we make use of IBM IoT Simulator which can transmit these parameters as required.
- This system uses a motion sensor to detect wild animals approaching near the field and smoke sensor to detect the fire. In such a case the sensor signals the microcontroller to take action.

1.2. PURPOSE

- An intelligent crop protection system helps the farmers in protecting the crop from the animals and birds which destroy the crop.
- This system also helps farmers to monitor the soil moisture levels in the field and also the temperature and humidity values near the field. The motors and sprinklers in the field can be controlled using the mobile application.

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

- Agriculture is a field which forms the basis of our economy. Yet it faces a lot of problems in terms of availability of resources, Irrigation, increasing rate of Pesticides, Climatic disasters, Insects which ruin the crops and makes a huge loss this sector.
- In agriculture water is needed for the crops for their growth. If the soil gets dry it is necessary to supply water. But sometime if the farmer doesn't visit the field it is not possible to know the condition of soil.
- Sometimes over supply of water or less supply of water affects the growth of crops.
- Sometimes if the weather/temperature changes suddenly it is necessary to take certain actions.
- Specific crops grow better in specific conditions, they may get damaged due to bad weather.

2.2 REFERENCES

1. Balaji Bhanu, Raghava Rao, J.V.N. Ramesh and Mohammed Ali hussain, “Agriculture Field Monitoring and Analysis using Wireless Sensor Networks for improving Crop Production”, Eleventh International Conference on Wireless and Optical Communications Networks (WOCN).2014.

2. HarshalMeharkure, Pargyline, Sheetal Israni, “Application of IOT Based System for Advance Agriculture in India”, International Journal of Innovative Research in Computer and Communication Engg(IJIRCCE) Vol.

3. Issue 11, pp. 10831-10837, 2015. 3. LIU Dan, Cao Xin, Huang Chongwei, JI Liang Liang, “Intelligent agent greenhouse environment monitoring system based on IOT technology”, International Conference on Intelligent Transportation, Big Data &Smart City, 2015.

4. Mehdi Roopei, Paul Rad, Kim-Kwang Raymond Choo, “Cloud of Things in smart agriculture: Intelligent irrigation monitoring by Thermal Imaging” IEEE CloudComputing,2017.

5. IBM cloud reference: https://cloud.ibm.com/6.
Python code reference: https://github.com/rachuriharish23/ibmsubscribe7.
IoT simulator: https://watson-iot-sensor-simulator.mybluemix.net/

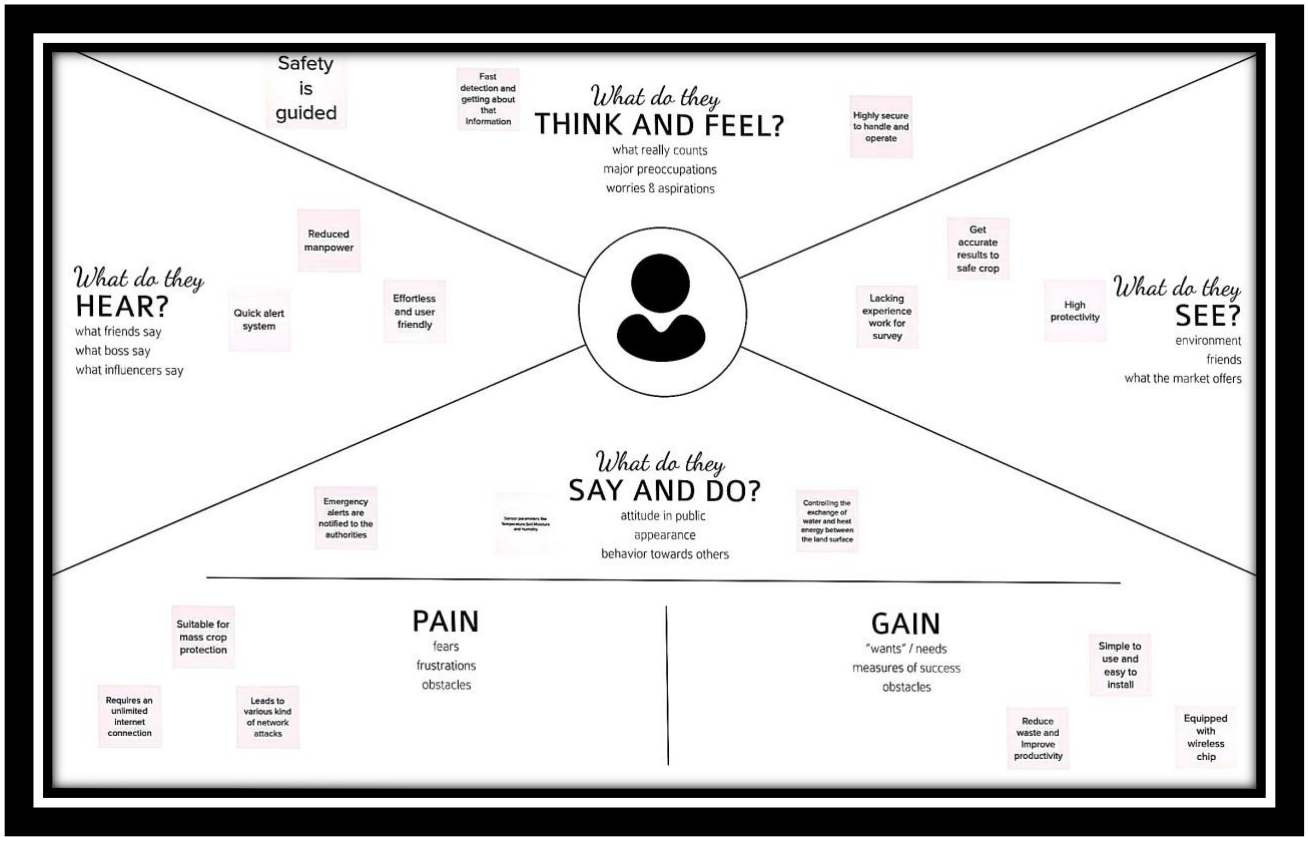
2.3. PROBLEM STATEMENT DEFINITION

- Smart Crop Protection System based on IoT can monitor soil moisture and climatic conditions to grow and yield a good crop.
- The farmer can also get the real time weather forecasting data by using external platforms like Open Weather API.
- Farmer is provided a mobile app using which he can monitor the temperature, humidity and soil moisture parameters along with weather forecasting details.
- Based on all the parameters he can water his crop by controlling the motors using the mobile application.
- Even if the farmer is not present near his field he can water his crop by controlling the motors using the mobile application from anywhere.
- Here we are using the Online IoT simulator for getting the Temperature, humidity and soil moisture values.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

IOT BASED SMART CROP PROTECTION SYSTEM FOR AGRICULTURE



An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviour and attitudes. It is a useful tool to helps teams better understand their users. Empathy mapping is a simple workshop activity that can be done with stakeholders, marketing and sales, product development, or creative teams to build empathy for end users.

3.2 BRAINSTORMING AND IDEATION

Brainstorming combines a relaxed, informal approach to problem solving with lateral thinking. It encourages people to come up with thoughts and ideas that can, at first, seem a bit crazy. Some of these ideas can be crafted into original, creative solutions to a problem, while others can spark even more ideas. This helps to get people unstuck by "jolting" them out of their normal ways of thinking.

IDEA 1:

- Crops in the farms are many times devastated by the wild as well as domestic animals and low productivity of crops is one of the reasons for this. It is not possible to stay 24 hours in the farm to sentinel the crops. So to surmount this issue an automated perspicacious crop aegis system is proposed utilizing Internet of Things (IOT).
- The system consists of esp8266 (nodeMCU), soil moisture sensor, dihydrogen monoxide sensor, GPRS and GSM module, servo motor, dihydrogen monoxide pump, etc. to obtain the required output. As soon as any critisicism is detected the system will engender an alarm to be taken and the lights will glow up implemented at every corner of the farm. This will not harm any animal and the crops will stay forfended.

IDEA 2:

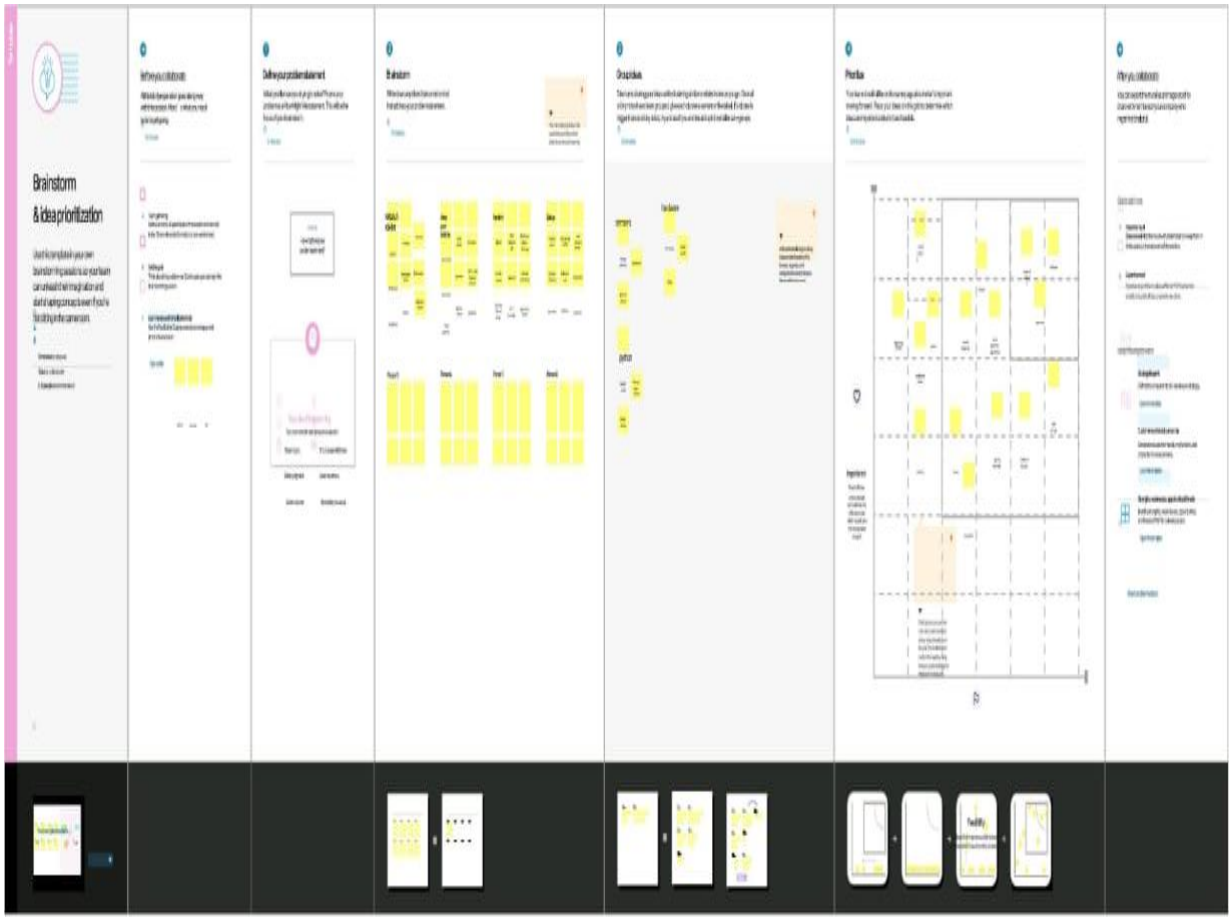
The Smart protection system defines that this project help to farmer for the protection of a farm. We have designed this project for the only secure from animals but this project have the provision to secure from the human begins also. This can be achieved by the help of IOT device. The SCPS work on the battery so that this project can be easily portable and also we are added solar

panels and converter modules. This can help the battery to charge from solar energy. The IOT device is used to indicate the farmer by a message while someone enter into the farm and we are used SD card module that helps to store a specified sound to fear the animals.

IDEA 3:

A centralizing method in the area of IIOT (Industrial Internet of Things) contrived for understanding agriculture which is preceding the arrangements low-power devices. This project yields a monitoring procedure for farm safety against animal attacks and climate change conditions. IIoT advances are frequently used in smart farming to emphasize the standard of agriculture. It contains types of sensors, controllers. On behalf of WSN, the ARM Cortex-A board which consumes 3W is the foremost essence of the procedure. Different sensors like DHT 11 Humidity & Temperature Sensor, PIR Sensor, LDR sensor, HC-SR04 Ultrasonic Sensor, and camera are mounted on the ARM Cortex-A board. The PIR goes high on noticing the movement within the scope, the camera starts to record, and the data will be reserved onboard and in the IoT cloud, instantaneously information will be generated automatically towards the recorded quantity using a SIM900A unit to notify about the interference with the information of the weather conditions attained by DHt11. If a variance happens, the announcement of the threshold rate will be sent to the cell number or to the website. The result will be generated on a catalog of the mobile of the person to take the necessary action.

BRAINSTORM



3.3 Proposed Solution

Attribute	Definition	Work
Novelty	The proposed solution needs to be fundamentally different from what people already know.	A toolbox of novel techniques based on the integration of crop prediction system and Internet of things.
Feasibility of idea	Project feasibility is the study of a project's various elements to determine if it has the potential for success	Earlier crop protection systems required manpower to detect intrusion and protect the crops. But, this project makes use of Iot technology with ultrasonic frequencies being used to prevent the intrusion
Business model	Create a model for identifying products and services to sell the market to target and also take into account anticipated expenses.	This project can be applied to different sectors of farming. Since it reduces the involvement of humans by bringing into picture new technologies, the cost of manufacturing also drops down. This makes the crop products easily available to the end user.
Social Impact	Social impact is how organizations, businesses or individuals' actions affect the surrounding community.	One of the major factors that has a direct impact on crop protection systems is the area where it is conceived. Different areas mean different types of crops and different types of intrusion. It therefore places a demand on the

		developer to configure the sensor values in such a way that it suits all kinds of environment and come up with a detection system that is common to all environments yet uniquely identifies the disturbance.
Scalability of solution	Scalability is system that can accommodate expansion without hampering the existing workflow and ensure an increase in the output or efficiency of the process.	Since this system uses computer vision techniques integrated with IBM cloudant services helps efficiently to retrieve images in large scale thus improving scalability.

3.4 Problem Solution Fit

1. Customer Segment Large scale Farmers and Soil owners.	2. Problems / Pains It is difficult for large scale farmers to manage and protect their resources from animal intrusions and external factors. There is also no specific software to manage and collect all the relevant information.	3. Triggers to act Real time water sprinklers for controlling humidity. Motion detectors to check on intruders and animals
4. Emotions Before: Stressed, Un prepared, Helpless After: Stress free, Fearless	5. Available Solutions Electric fences. Humidity Management Models. Crop Management Software.	6. Customer Limitations Animal Intrusions. Effects due to environment. Fertility of soil
7. Behaviour Gain knowledge on the existing solutions and try to learn more on the products available inthis domain.	8.Channels of Behavior Gather informationfrom websites and journals about the existing models.	9. Problem root / Cause Wild Animals. Environmental Factors. (Excess greenhouse gasses, High Temperatures) Soil fertility.
10Your Solution ● Crop protection from animals using IR motion detectors ● A user interface system for farmers to analyze the data ● Crop protection from environmental factors such as UV rays, temperature, humidity, moisture content in soil		

4. REQUIREMENT ANALYSIS

4.1. Functional requirement

FR.NO.	**Functional requirements (Epic no)**	**Sub Requirements (Story/sub task)**
FR-1	User registration	Download the app Registration through email Create an account Follow the instructions.
FR-2	User confirmation	Confirmation via Email Confirmation via OTP
FR-3	Interface sensor	Interface sensor and the application so if animals enter the field it gives alarm.
FR-4	Accessing datasets	Datasets are retrieved from Cloudant DB
FR-5	Mobile application	Motors and sprinklers in the field can be controlled by mobile application.

4.2 Non functional requirements

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	The smart protection system defines that this project helps farmers to protect the farm.
NFR-2	Security	We have designed this project to secure the crops from animals.
NFR-3	Reliability	This project will help farmers in protecting their fields and save them from significant financial losses. This will also help them in achieving better crop yields thus leading to their economic well being.
NFR-4	Performance	IOT devices and sensors are used to indicate the farmer by a message when animals try to enter into the field and also we use an SD card module that helps to store a specified sound to scare the animals.
NFR-5	Availability	By developing and deploying resilient hardware and software we can protect the crops from wild animals.
NFR-6	Scalability	Since this system uses computer vision techniques integrated with IBM cloudant services helps efficiently to retrieve images in large scale thus improving scalability.

5. PROJECT DESIGN

5.1. Data Flow Diagrams

A data flow diagram (DFD) maps out the flow of information for any process or system.It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyze an existing system or model a new one. Like all the best diagrams and charts, a DFD can often visually "say" things that would be hard to explain in words, and they work for both technical and nontechnical audiences, from developer to CEO. That's why DFDs remain so popular after allthese years. While they work well for data flow software and systems, they are less applicable nowadays to visualizing interactive, real-time or database-oriented software or systems.

There are four main elements of a DFD — external entity, process, data store, and data flow.

External entity

An external entity, which are also known as terminators, sources, sinks, or actors, are an outside system or process that sends or receives data to and from the diagrammed system. They're either the sources or destinations of information, so they're usually placed on the diagram's edges. External entity symbols are similar across models except for Unified, which uses a stick-figure drawing instead of a rectangle, circle, or square.

Process

Process is a procedure that manipulates the data and its flow by taking incoming data, changing it, and producing an output with it. A process can do this

by performing computationsand using logic to sort the data, or change its flow of direction. Processes usually start from thetop left of the DFD and finish on the bottom right of the diagram.

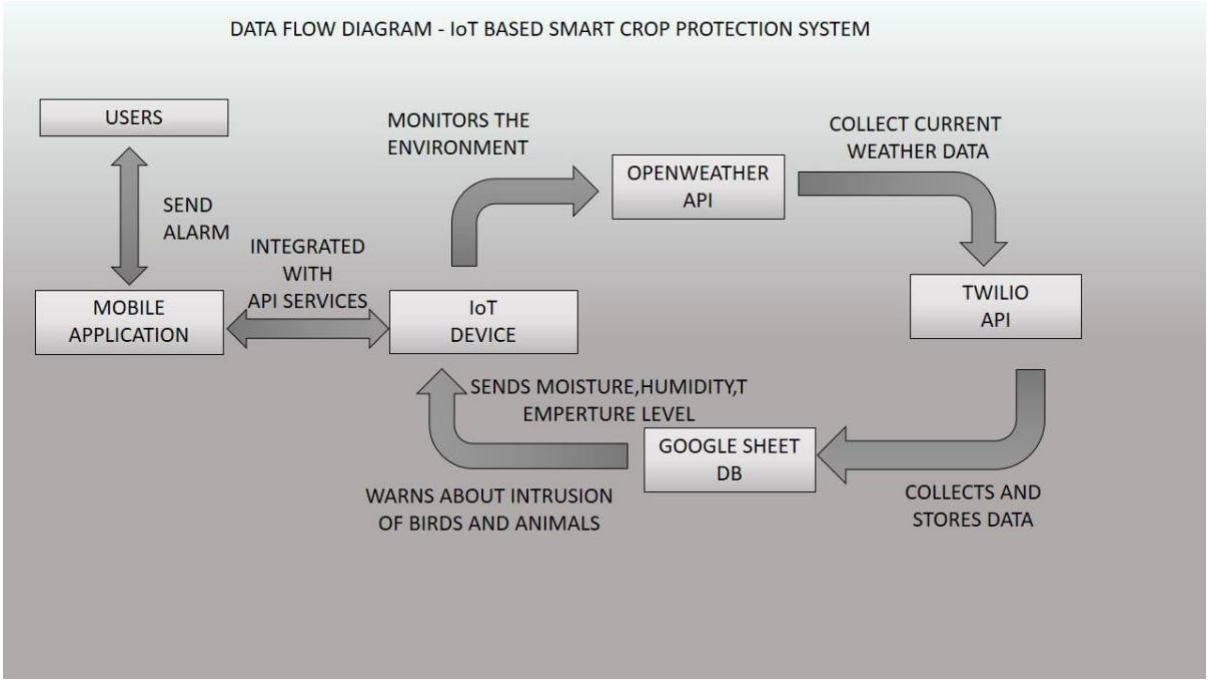
Data store

Data stores hold information for later use, like a file of documents that's waiting to be processed. Data inputs flow through a process and then through a data store while data outputsflow out of a data store and then through a process.

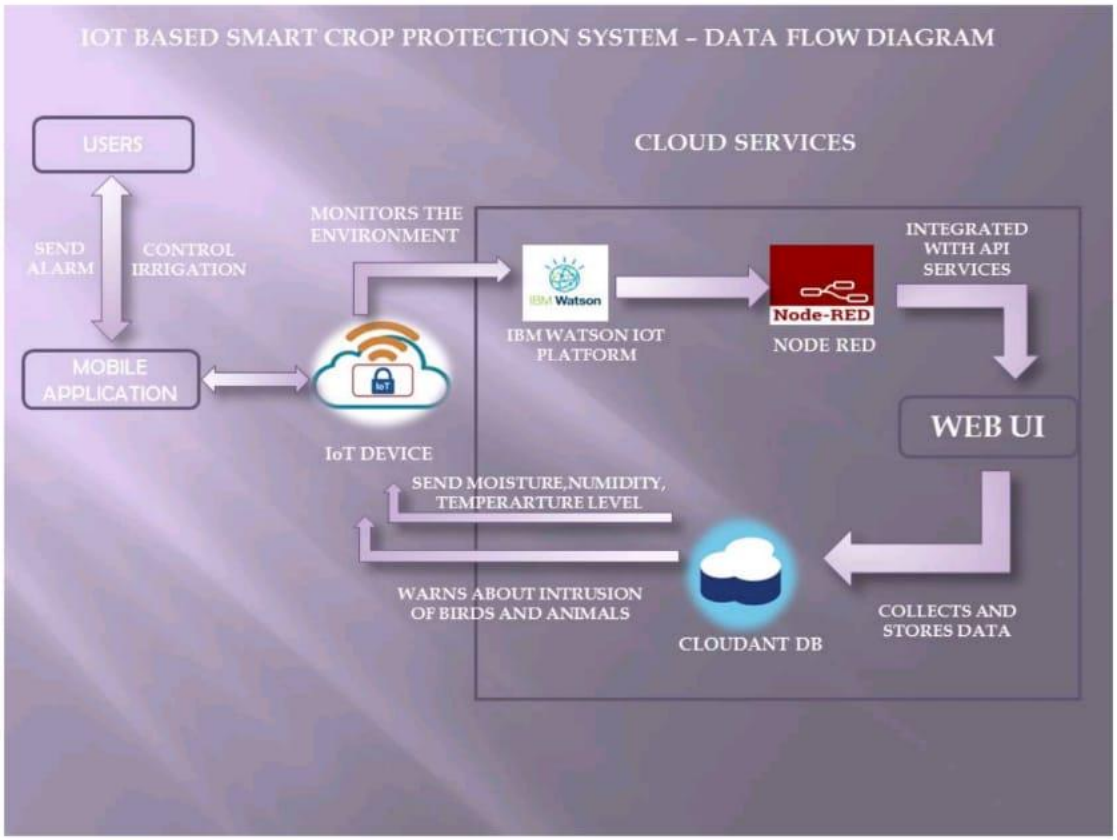
Data flow

Data flow is the path the system's information takes from external entities through processes and data stores. With arrows and succinct labels, the DFD can show the direction ofthe data flow.

The data flow diagram for IOT based smart crop protection system using for agriculture isshown in following figure 5.1

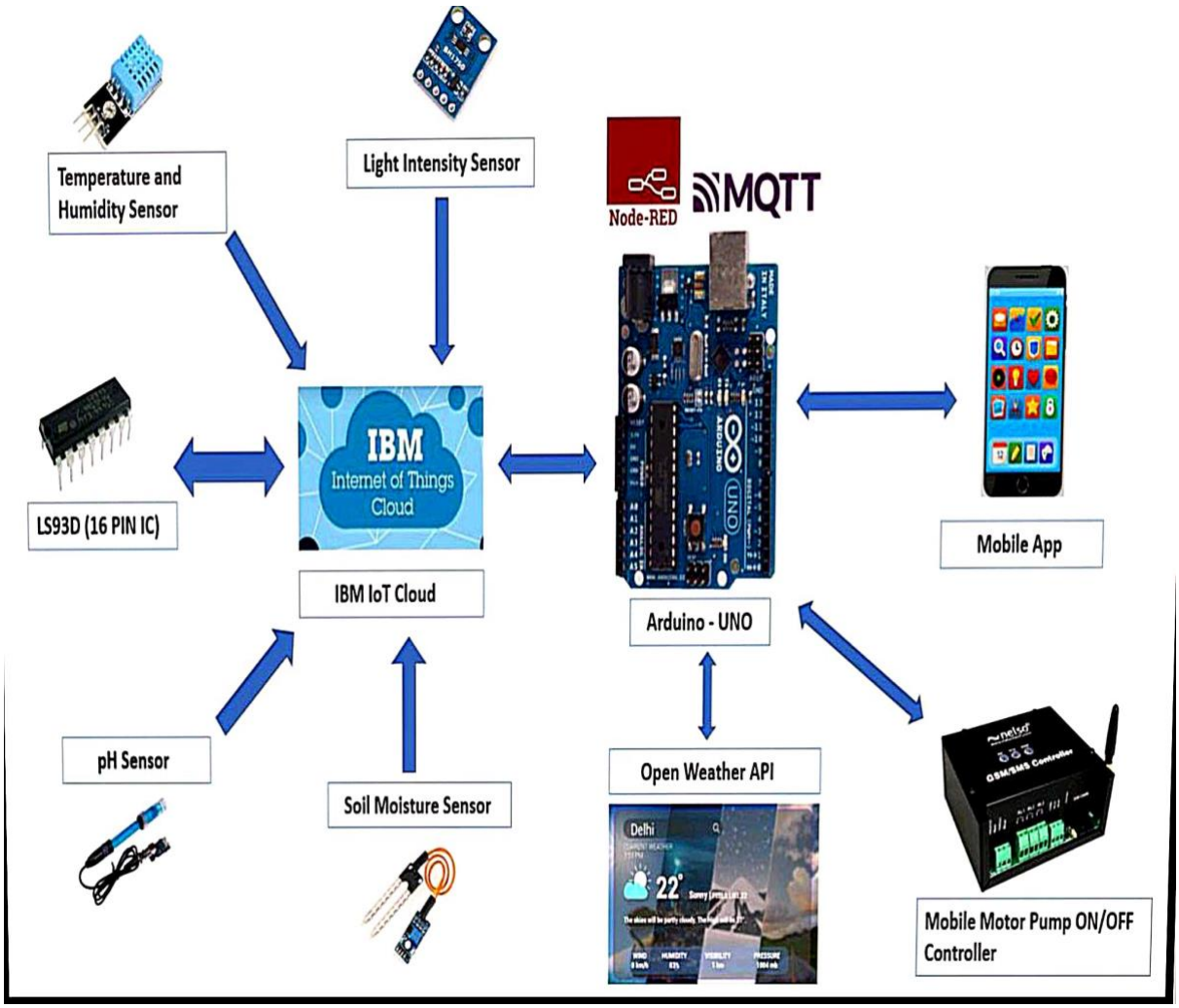


Data Flow Diagram for IoT based smart crop protection system


IOT BASED SMART CROP PROTECTION SYSTEM – DATA FLOW DIAGRAM
USERS
CLOUD SERVICES
SEND ALARM
CONTROL IRRIGATION
MONITORS THE ENVIRONMENT
IBM Watson
IBM WATSON IOT PLATFORM
Node-RED
NODE RED
INTEGRATED WITH API SERVICES
MOBILE APPLICATION
IoT
IoT DEVICE
WEB UI
SEND MOISTURE,NUMIDITY, TEMPERARTURE LEVEL
WARNS ABOUT INTRUSION OF BIRDS AND ANIMALS
CLOUDANT DB
COLLECTS AND STORES DATA

5.2. Solution & Technical architecture

Technical Architecture is the name of the total concept that is applied to the IT Infrastructure of an organization. IT Infrastructure is a coherent set of interconnected hardware and software, like networks, clouds, servers, clients, printers, tablet PC, smart phones.



6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint planning is an event in scrum that kicks off the sprint. The purpose of sprint planning is to define what can be delivered in the sprint and how that work will be achieved. Sprint planning is done in collaboration with the whole scrum team.

The sprint is a set period of time where all the work is done. However, before leap into action it is necessary to set up the sprint. It need to decide on how long the time box is going to be, the sprint goal, and where it is going to start. The sprint planning session kicks off the sprint by setting the agenda and focus. If done correctly, it also creates an environment where the team is motivated, challenged, and can be successful.

The Table shows the sprint planning and estimation of IoT Based Smart Crop Protection System.

6.2 Sprint Delivery Schedule

SPRINT	TOTAL STORY POINTS	DURATION	SPRINT START DATE	SPRINT END DATE (PLANNED)	STORY POINTS COMPLETED (AS ON PLANNED END DATE)	SPRINT RELEASE DATE (ACTUAL)
Sprint-1	20	6 Days	26 Oct 2022	29 Oct 2022	20	01 Nov2022
Sprint-2	20	6 Days	01 Nov 2022	05 Nov 2022	20	07 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	13 Nov 2022	20	14 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

7. CODING &SOLUTIONING

7.1 Feature1

- Warning and text message to mobile number when animal is found in the farm
- Online database using Google Sheets to store the results of the sensors
- Using weather API to notify if the weather is extreme to the user
- Soil moisture is also monitored and notified to the user through SMS about it, so that thefarmer can ensure the required level of moisture in the soil.

Basic explanation:

Libraries required:

```
In [6]: import pandas as pd
        import random
        import requests, json
        import time
        from twilio.rest import Client
        import gspread
        from datetime import datetime
```

Pandas :

Pandas is an open source Python package that is most widely used for data science/data analysis and machine learning tasks. It is built on top of another package namedNumpy, which provides support for multi-dimensional arrays

Random:

Sometimes we want the computer to pick a random number in a given range, pick arandom element from a list, pick a random card from a deck, flip a coin, etc. The random module provides access to functions that support these types of operations.

In this case random is used to generate the random values between specified float values torecreate sensor data.

JSON:

JSON is an open standard file format and data interchange format that uses human- readable text to store and transmit data objects consisting of attribute–value pairs and arrays. Itis a common data format with diverse uses in electronic data interchange, including that of webapplications with servers.

Time:

Python time module allows to work with time in Python. It allows functionality like getting the current time, pausing the Program from executing, etc. In this program the time library is used to get the current time so that the data can be stored using time stamps.

Creating Random data

```
In [3]: rand_prox_location_1=[]
        constrain_location_1=9.7
        rand_prox_location_2=[]
        constrain_location_2=7
        rand_prox_location_3=[]
        constrain_location_3=9.5
        rand_prox_location_4=[]
        constrain_location_4=9.9
        soil_humidity=[]
        constrain_soil_humidity= 75
        constrain_temp= 295
        constrain_humidity= 80
        constrain_pressure= 1030
        constrain_soil_humidity_low =40
        for i in range(1,100):
            rand_prox_location_1.append(round(random.uniform(0.25,10),2))
            rand_prox_location_2.append(round(random.uniform(0.25,10),2))
            rand_prox_location_3.append(round(random.uniform(0.25,10),2))
            rand_prox_location_4.append(round(random.uniform(0.25,10),2))
            soil_humidity.append(round(random.uniform(1,100),1))
        print("--------",rand_prox_location_4)
        print("--------",rand_prox_location_3)
        print("--------",rand_prox_location_2)
        print("--------",rand_prox_location_1)
        print("--------",soil_humidity)

        -------- [5.21, 7.96, 3.89, 1.73, 5.69, 4.45, 8.86, 1.75, 3.76, 8.3, 7.96, 2.56, 5.3, 1.86, 8.64, 8.3
```

In this particular snippet a data frame ‘df’ is created

And rand_prox_location_1 to ran_rpox_location_4 lists are created so that the random values can be temporarily stored in the runtime. These lists represent the various PIR sensor outputs which are in meters (distance from the pir sensor), constrain_location_1 to constrain_location_4 are the constraints provided for each sensor, which can be changed so that the values for each and every field(farm fields) dimensions.

Similarly soil humidity list and constraints are also created to represent the sensor data from the soil humidity sensor, constraints for the temperature, atmospheric humidity, pressure werealso assigned.

In the following random uniform float values are generated and appended in the respective lists.

```
In [69]: df['Location_1']=rand_prox_location_1
         df['Location_2']=rand_prox_location_2
         df['Location_3']=rand_prox_location_3
         df['Location_4']=rand_prox_location_4
         df['soil_humidity']=soil_humidity
         df.to_excel('IBM_sensor_data.xlsx', index = False)
```

The location data and soil humidity data were then stored in an excel file knows as IBMV sensor data for further use.

program for temperatures

```
In [73]: for i in range(100):
             time_stamp = datetime.now()
             time.sleep(5)
             temp=[0,0,0,0]
             value=[0,0,0,0]
             animal_intrusion_set=0
             temp_set=0
             soil_humidity_set=0
             pressure_set=0
             atmos_humidity_set=0
             animal_intrusion_text = ' Animal intrusion at'
             temporary_text= ''
             current_temperature,current_pressure,current_humidity,weather_description = get_whether()
             if current_temperature > constrain_temp:
                 temp_set=1
                 temporary_text += ' Temperature is high :'+str(current_temperature) + ' || '
             if current_pressure > constrain_pressure:
                 pressure_set=1
                 temporary_text += ' Pressure is high :'+str(current_pressure) + ' || '
             if current_humidity > constrain_humidity:
                 atmos_humidity_set=1
                 temporary_text += ' Atmosphereic humidity is high :'+str(current_humidity ) + ' || '
```

FEATURE-2:

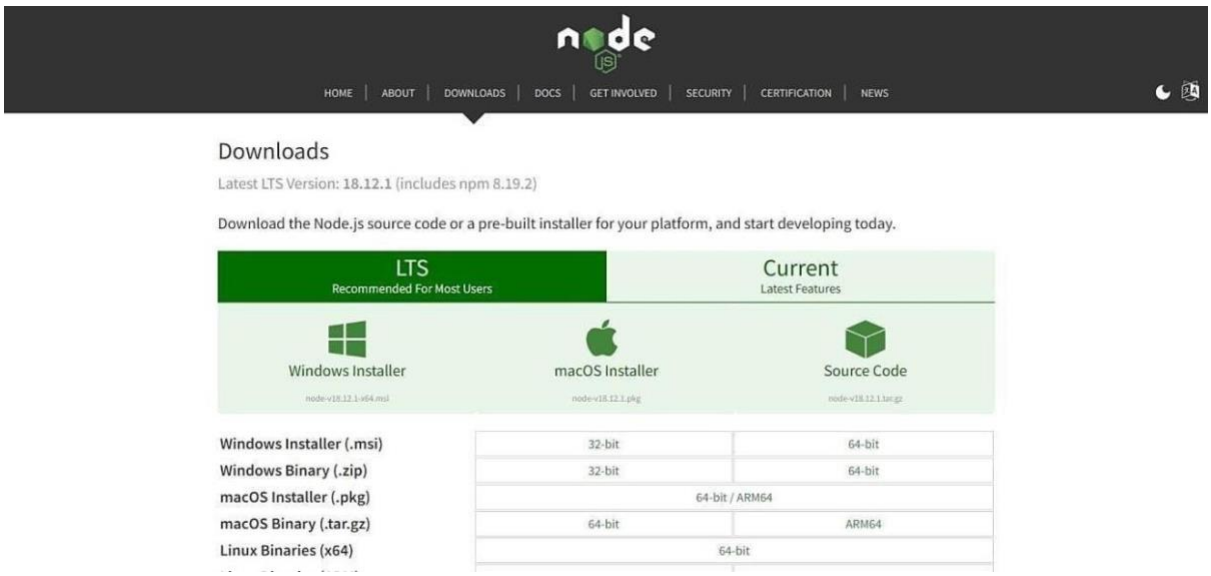
- Good sensitivity to combustable gas in wide range.
- High sensitivity to LPG, Propane and Hydrogen.
- Long life and low cost. iv. Simple drive circuit.

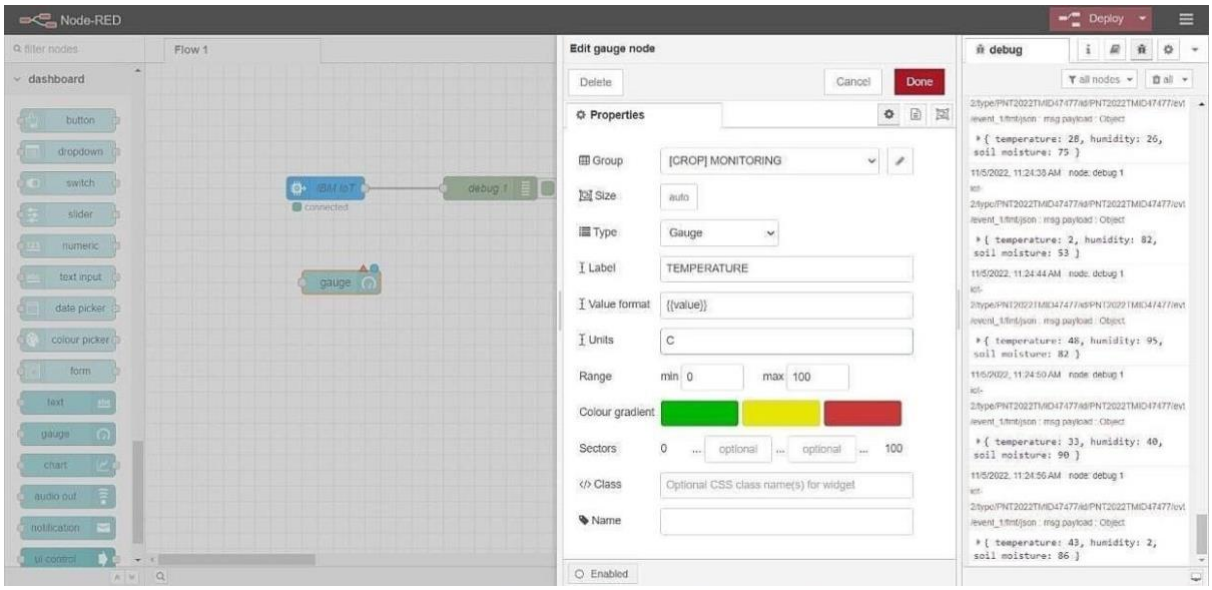
8.TESTING

8.1TEST CASES:

S.No	parameter	Values	Screenshot
1	Model summary	-	
2	accuracy	Training accuracy- 95% Validation accuracy- 72%	
3	Confidence score	Class detected- 80% Confidencescore- 80%	

8.2 User Acceptance Testing:




Node-RED
Deploy
Flow 1
dashboard
button
dropdown
switch
slider
numeric
text input
date picker
colour picker
form
text
gauge
chart
audio out
notification
ui control
IBM IoT
connected
debug 1
gauge
Edit gauge node
Delete
Cancel
Done
Properties
Group
[CROP] MONITORING
Size
auto
Type
Gauge
Label
TEMPERATURE
Value format
{{value}}
Units
C
Range
min 0
max 100
Colour gradient
Sectors
0
optional
optional
100
Class
Optional CSS class name(s) for widget
Name
Enabled
debug
all nodes
all
{ temperature: 28, humidity: 26, soil moisture: 75 }
11/5/2022, 11:24:38 AM node: debug 1
{ temperature: 2, humidity: 82, soil moisture: 53 }
11/5/2022, 11:24:44 AM node: debug 1
{ temperature: 48, humidity: 95, soil moisture: 82 }
11/5/2022, 11:24:50 AM node: debug 1
{ temperature: 33, humidity: 40, soil moisture: 90 }
11/5/2022, 11:24:56 AM node: debug 1
{ temperature: 43, humidity: 2, soil moisture: 86 }

9.RESULTS

- The problem of crop vandalization by wild animals and fire has become a major socialproblem in current time.

- It requires urgent attention as no effective solution exists till date for this problem.

- Thus this project carries a great social relevance as it aims to address this problem.

- This project will help farmers in protecting their orchards and fields and save them from significant financial losses and will save them from the unproductive efforts thatthey endure for the protection their fields.

- This will also help them in achieving bettercrop yields thus leading to the economic wellbeing.

10.ADVANTAGES AND DISADVANTAGES

ADVANTAGES

This proposed work successfully designs a working prototype of an integrated IoT hardware with an platforms like Twilio that fulfils the following functions of timer-based irrigation control, real-time monitoring of farm data and prediction of the weather condition ofthe crops.

It also integrated with the function of warning the activity of animals and birds in the fields to the users. The real-time data from the farm such as soil moisture are collected from the environment using the sensors that is interfaced respectively.

Another novel feature added in this model is weather API that notifies the weather condition when it goes beyond the expected weather condition. It also comes up with the usageof Twilio ,a platform which is provides programmable communication tools for making and receiving phone calls ,sending and receiving messages and performing other communication functions using its web service APIs.

Here a message is sent to the user when animal is found in the crop field. This proposedprototype system uses the Open Weather map API which provides the real time weather condition for any geographical location.

This system uses the Google Sheets which acts as a Database system that collects the sensor data and store it for future reference to the users.

DISADVANTAGES

The use of technology in farming and agriculture making it smart agriculture, is of course, a good initiative and a much-needed one with the present increasing demand in the food supply.But there is the chance where this proposed smart farming protection system will require certain skill sets in particular in order to understand and operate the equipment. In the case of equipment computer-based intelligence for running the devices, it is highly unlikely that a normal farmer will be able to possess this knowledge or even develop them.

Farmers are not used to these high-end technologies. They do not understand computer language or the artificial intelligence. For the smart agriculture, Internet of Things is essential which will require artificial intelligence and computer-based intelligence.

This cannot be balanced here. To overcome this challenge, the devices will have to be changed in a dramatic fashion so as to make it understandable for farmers.

This also means that the devices should be somewhere in between where the technology experts and farmers can both communicate about it.

And also this system needs availability of internet continuously. Rural part of most of the developing countries do not fulfil this requirement. Moreover internet connection is slower.

11. Conclusion

Smart farming is a modern farming management concept with IoT technology to increase the productivity in agriculture. With the use of smart farming, users can effectively monitor the crop field the quality and quantity of their crops. Users cannot be physically present on the field 24 hours a day. In addition, the farmersmay not have the knowledge to use different tools to measure the ideal environmentalconditions for their crops. IoT provides them with the automated system, which can function without any humansupervision and can notify them to make proper decision to deal with different kind of problemsthey may face during farming.

It has the capability to reach and notify the farmer even if farmer is not on the field, which can allow farmer to manage more farmland, thus improving their production.Thus, we can conclude that this IoT based smart crop protection system will definitelyhelp users in farmland to effectively monitor their crops with the user-friendly platforms and other alert means.

12.FUTURE SCOPE

- The proposed work system is a successful working prototype that fulfils to protect cropsfrom the intrusion of animals and birds.
- This system will helps the users to monitor the temperature and to notify the weather conditions.
- This system assuredly assists the users to know about the soil moisture level. And the IoT based smart crop protection system implemented here brings a naval approach crop protection system from animals.
- This assures the early detection and prevention of incurring losses due to the damage of crops.
- The following suggestions may be carried out in future implementation of the system; the smart crop prediction may be also carried out by considering the various factors like NPKcontent of the soil, UV radiation along with the tracking of the crop field location using GPS module system. The automated pest traps also be introduced using image recognition techniques and neural networks in smart protection system

13.APPENDIX

Source code

Python script

```
import cv2

import numpy as np

import wiotp.stk.device

import playsound

import random

import time

import datatime

import ibm_boto3

from ibm_botocore.client import config,ClientError

#cloudant DB

from cloudant.client import Cloudant

from cloudant.error import CloudantException

from cloudant.result import Result, ResultByKey

from clarifai_grpc.channel.clarifai_channel import ClarifaiChannel

from clarifai_grpc.grpc.api import service_pb2_grpc

stub=service_pb2_grpc.V2stub(ClarifaiChannel.get_grpc_channel())

from clarifai_grpc.grpc.api import service_pb2, resources_pb2

from clarifai_grpc.grpc.api.status import status_code_pb2

metadata=(('authorization','key 98753f449e380e53c2edacec0dbc0c53'))#clarifai service credentials

COS_ENDPOINT="s3.tok.ap.cloud-object-storage.appdomain.cloud"

COS_API_KEY_ID="v75mpBmsCjMwg_iPwSHWnwAp0lK2gDuagAegza9DbseY"

COS_AUTH_ENDPOINT="https://control.cloud-object-storage.cloud.ibm.com/v2/endpoints"
```

```
COS_RESOURCE_CRN="crn:v1:bluemix:public:cloud-object-storage:global:a/bdbbdf21a1d84d54b194bf209872657d:1677a877-cc75-4eba-a0a1-9a4cbd6de882::"
clientdb=Cloudant(_)
clientdb.connect()
#createresource
cos=ibm_boto3.resource("s3",
    ibm_api_key_id=COS_API_KEY_ID,
    ibm_service_instance_id=COS_RESOURCE_CRN,
    ibm_auth_endpoint=COS_AUTH_ENDPOINT,
    config=config(signature_version="oauth"),
    endpoint_url=COS_ENDPOINT
)
def multi_part_upload(bucket_name,item_name,file_path):
    try:
        print("Starting file transfer for {0}to bucket:{1}\n".format(item_name,bucket_name))
        part_size=1024*1024*5
        file_threshold=1024*1024*15
        transfer_config=ibm_boto3.s3.transfer.TransferConfig(
            multipart_threshold=file_threshold,
            multipart_chunksize=part_size
            )
        with open(file_path,"rb") as file_data:
                cos.Object(bucket_name,item_name).upload_fileobj(
                    Fileobj=file_data,
                    Config=transfer_config
                    )
```

```
        print ("Transfer for {0} complete!\n".format(item_name))
    except ClientError as be:
        print ("CLIENT ERROR:{0}\n".format(be))
    except Exception as e:
         print ("Unable to complete multi-part upload:{0}".format(e))

def myCommandCallback(cmd):
    print("Command received:%s"%cmd.data)
    command=cmd.data['command']
    print(command)
    if(command=='lighton'):
        print('lighton')
    elif(command=='lightoff'):
        print('lightoff')
    elif(command=='motoron'):
        print('motoron')
    elif(command=='motoroff'):
        print('motoroff')
myConfig={
    "identity":{
        "orgId":"kc06ni",
        "typeId":"abcd",
        "deviceId":"123"
    },
    "auth":{
        "token":"12345678"
    }
```

```
}
client=wlotp.sdk.device.DeviceClient(config=myconfig, logHandlers=None)
client.connect()

database_name="sample1"
my_database=clientdb.create_database(database_name)
if my_database.exists():
    print(f" '{database_name}' successfully created.")
    cap=cv2.VideoCapture("c:\python\python37\garden.mp4.mp4")
    if(cap.isopened()==True):
        print('file opened')
    else:
        print('file not found')
while(cap.isopened()):
    ret,frame=cap.read()
    gray=cv2.cvtColour(frame,cv2.COLOUR_BGR2GRAY)
    ims=cv2.resize(frame,(960,540))
    cv2.imright('ex.jpg',ims)
    with open("ex,.jpg","rb") as f:
        file_bytes=f.read()
    request=service_pb2.PostMode1OutputsRequest(
        model_id='general-image-recognition',

inputs=[resources_pb2.Input(data=resources_pb2.Data(image=resources_pb2.I
mage(base64=file_bytes)))])
    response=stub.PostMode1Outputs(request,metadata=metadata)
    if response.status.code!=status_code_pb2.SUCCESS:
        raise Exception("Request failed,status code:"+str(response.status.code))
```

```
    detect=False
    for concept in response.outputs[0].data.concepts:
        if(concept.value>0.98):
            if(concept.name=="animal"):
                print("Alert!Alert! Animal detected")
                #playsound
                picname=datetime.datetime.now().strftime("%y-%m-%d-%H-%M")
                cv2.imwrite(picname+'.jpg',frame)
                multi_part_upload('joni',picname+'.jpg',picname+'.jpg')

json_document={"link":COS_ENDPOINT+'/'+'joni'+'/'+picname+'.jpg'}
                new_document=my_database.create_document(json_document)
                if new_document.exists():
                   print(f"Document successfully created.")
                time.sleep(5)
                detect=True
    moist=random.randint(0,100)
    temp=random.randint(0,100)
    humidity=random.randint(0,100)
    myData={'Animal':
detect,'temperature':temp,'moisture':moist,'humidity':humidity}
    print(myData)
    if (humidity!=None):

client.publishEvent(eventId="status",msgFormat="json",data=myData,qos=0,on
Publish=None)
        print("Publish ok...")
    client.commandCallback=myCommandCallback
    cv2.imshow('frame',ims)
```

```
    if cv2.waitKey(1) & 0xFF==ord('q'):
        break
client.disconnect()
cap.release()
cv2.destroyAllWindows()SS
```